

IN THE CLAIMS

Please amend claims 11, 13, 20, 25, and 26 as indicated below.

1. (Original) A microprocessor comprising:

an instruction buffer, wherein said instruction buffer is configured to store a plurality of instructions;

a load prediction unit coupled to said instruction buffer, wherein said load prediction unit is configured to:

detect a first load instruction of said plurality of instructions;

predict a first load address of said first load instruction; and

identify a first instruction of a new thread; and

a data cache coupled to said load prediction unit, wherein said data cache is configured to:

receive said first load address; and

fetch data corresponding to said first load address in response to detecting said data is not present in said data cache.

2. (Original) The microprocessor of claim 1, wherein said load prediction unit comprises:

a load prediction table with a plurality of entries; and

circuitry which supports load address prediction and new thread creation.

3. (Original) The microprocessor of claim 2, wherein said circuitry is configured to detect said first load instruction by scanning said plurality of instructions in said instruction buffer for instructions with opcodes which correspond to load instructions.

4. (Original) The microprocessor of claim 2, wherein said plurality of entries in said load prediction table comprises a valid field, an instruction address field, and an effective address field.
5. (Original) The microprocessor of claim 4, wherein said load prediction table further comprises a stride field and a threshold field.
6. (Original) The microprocessor of claim 4, wherein said load prediction unit is configured to create a first entry in said load prediction table for said first load instruction by storing an instruction address in an instruction address field of an entry and storing an effective address in an effective address field of said entry and storing a value in a valid field of said entry which indicates said entry is valid.
7. (Original) The microprocessor of claim 6, wherein a stride value is stored in a stride field of said entry and a threshold field of said entry is initialized to indicate no misses for said first load instruction have been recorded.
8. (Original) The microprocessor of claim 6, wherein said load prediction unit is configured to detect said first entry in said load prediction table by comparing an instruction address of said first detected load instruction to addresses in instruction address fields of said plurality of entries, wherein said instruction address of said first detected load instruction corresponds to said instruction address stored in said instruction address field of said first entry, wherein said valid field of said first entry indicates said entry is valid.
9. (Original) The microprocessor of claim 6, wherein said load prediction unit is configured to predict a load address of said first detected load instruction by adding a first contents of an effective address field of said first entry to a second contents of a stride field of said first entry.

10. (Original) The microprocessor of claim 7, wherein said load prediction unit is configured to update an entry of said first detected load instruction of said load prediction table by storing a difference between a received effective address and said contents of said effective address field in said stride field and by storing said received effective address in said effective address field, in response to detecting said detected load instruction hit in said data cache.
11. (Currently Amended) The microprocessor of claim 7, wherein said load prediction unit is configured to update an entry of said first detected load instruction of said load prediction table by storing a difference between a received effective address and said contents of said effective address field in said stride field and by storing said received effective address in said effective address field and by incrementing [[a]] contents of a threshold field of said first entry, in response to detecting said detected load instruction missed in said data cache.
12. (Original) The microprocessor of claim 5, wherein said load prediction unit is further configured to predict a load instruction will miss.
13. (Currently Amended) The microprocessor of claim 5, wherein said load prediction unit is configured to predict said first detected load instruction will miss, in response to detecting [[a]] contents of a threshold field of said first entry equals a threshold value.
14. (Original) The microprocessor of claim 13, wherein in response to predicting said first detected load instruction will miss, said load prediction unit is configured to identify a first instruction of a new thread.
15. (Original) The microprocessor of claim 14, wherein said first instruction is said identified by comparing a destination register of said first detected load instruction with a destination register of instructions in said instruction buffer, wherein said destination register of said first instruction is the same register as said destination

register of said first detected load instruction, wherein said instructions in said instruction buffer are subsequent in program order to said first detected load instruction.

16. (Original) The microprocessor of claim 15, wherein said first instruction is a load instruction.
17. (Original) The microprocessor of claim 14, wherein said first instruction is said identified by detecting said first instruction is an unconditional branch.
18. (Original) The microprocessor of claim 14, wherein said first instruction is said identified by detecting said first instruction immediately follows a loop iteration branch.
19. (Original) A method of load address prediction and thread instruction identification is contemplated, said method comprising:
 - detecting a first instruction of a plurality of instructions in an instruction buffer is a load instruction;
 - predicting a load address for said first instruction, in response to detecting a valid entry exists in a load prediction table for said first instruction;
 - updating said entry in said load prediction table;
 - predicting said first instruction will miss, in response to detecting a miss threshold condition has been met; and
 - identifying a first thread instruction of a new thread, in response to said predicting said first instruction will said miss.

20. (Currently Amended) The method of claim 19, wherein said detecting a first instruction of a plurality of instructions in an instruction buffer is a load instruction comprises scanning said plurality of instructions for instructions with ~~op-codes~~ opcodes corresponding to load instructions.
21. (Original) The method of claim 19, further comprising determining if a valid entry exists in a load prediction table for said first instruction, wherein said determining comprises comparing an instruction address of said first instruction to addresses in instruction address fields of said load prediction table, wherein said instruction address of said first instruction corresponds to said instruction address stored in said instruction address field of said valid entry, wherein said valid field of said valid entry indicates said entry is valid.
22. (Original) The method of claim 19, further comprising creating an entry in said load prediction table for said first instruction, in response to detecting no valid entry corresponding to said first instruction exists, wherein said creating comprises:
- storing an instruction address in an instruction address field of an entry;
 - storing an effective address in an effective address field of said entry; and
 - storing a value in a valid field of said entry, wherein said value indicates said entry is valid.
23. (Original) The method of claim 22, further comprising:
- storing a stride value in a stride field of said entry; and
 - initializing a threshold field of said entry to indicate no misses for said first instruction have been recorded.

24. (Original) The method of claim 19, further comprising computing a predicted address for said first instruction, in response to detecting said valid entry does exists, wherein said computing comprises adding a first contents of an effective address field of said valid entry to a second contents of a stride field of said valid entry.

25. (Currently Amended) The method of claim 19, wherein said updating comprises:

storing a difference between a received effective address and $[[a]]$ contents of an effective address field of said entry, wherein said difference is stored in a stride field of said entry;

storing said received effective address in said effective address field of said entry;
and

incrementing $[[a]]$ contents of a threshold field of said first entry, in response to detecting a valid entry exists in said load prediction table for said first instruction and said first instruction missed.

26. (Currently Amended) The method of claim 19, wherein said updating comprises:

storing a difference between a received effective address and $[[a]]$ contents of an effective address field of said entry, wherein said difference is stored in a stride field of said entry; and

storing a received effective address in said effective address field, in response to detecting a valid entry exists in said load prediction table for said first instruction and said first instruction hit.

27. (Original) The method of claim 19, wherein said predicting said first load instruction will miss comprises detecting a value in a threshold field of said entry equals a threshold value.

28. (Original) The method of claim 19, wherein said identifying said first thread instruction of said new thread comprises:

comparing a destination register of said first instruction with a destination register of a second plurality of instructions in said instruction buffer;

detecting a destination register of a second instruction of said second plurality of instructions is the same register as said destination register of said first instruction; and

selecting said second instruction as said first thread instruction, wherein said second plurality of instructions in said instruction buffer are subsequent in program order to said first instruction.

29. (Original) The method of claim 28, wherein said first thread instruction is a load instruction.

30. (Original) The method of claim 19, wherein said identifying said first thread instruction comprises:

detecting a second instruction of a second plurality of instructions in said instruction buffer is an unconditional branch; and

selecting said second instruction as said first thread instruction, wherein said second plurality of instructions in said instruction buffer are subsequent in program order to said first instruction.

31. (Original) The method of claim 19, wherein said identifying said first thread instruction comprises:

detecting a second instruction of a second plurality of instructions in said instruction buffer immediately follows a loop iteration branch; and

selecting said second instruction as said first thread instruction, wherein said second plurality of instructions in said instruction buffer are subsequent in program order to said first instruction.